VISIT…

LANZAROTE
Caliente.COM

Материалы XI Всероссийского семинара "Нейроинформатика и ее приложения", Красноярск, 2003. - 215с. - С.171-175.

ВЗГЛЯД НА АРХИТЕКТУРУ И ТРЕБОВАНИЯ К НЕЙРОИМИТАТОРУ ДЛЯ РЕШЕНИЯ СОВРЕМЕННЫХ ИНДУСТРИАЛЬНЫХ ЗАДАЧ

Царегородцев В.Г.
www.NeuroPro.ru tsar@neuropro.ru

Рассмотрен опыт автора по созданию и использованию программы-нейроимитатора для решения широкого круга задач прогнозирования и классификации на базе наборов экспериментальных данных. Дано описание идей организации внутренней архитектуры программы и указаны реализованные нейросетевые, статистические и эмпирические методы обработки данных, составляющий авторский "ритуал" анализа данных.

Нейросетевые методы анализа и обработки данных в современной практике

Для решения сложных неформализованных задач прогнозирования и классификации широко применяются искусственные обучаемые нейронные сети [1,2]. Несколько основных нейросетевых архитектур, такие, как многослойные персептроны, сети и карты Кохонена, делают возможным решение широкого спектра задач, зачастую нерешаемых классическими статистическими методами обработки данных. Среди привлекающих пользователя достоинств нейронных сетей можно выделить такие:

- Обучаемость на наборе примеров.
- Построение нелинейной регрессионной зависимости или нелинейной разделяющей поверхности без априорного задания вида нелинейной функции с точностью до значений параметров, идентифицируемых в дальнейшем.
- Возможность решения одновременно нескольких задач прогнозирования или классификации одной нейромоделью с векторным выходом.
- Целевая функция, оптимизируемая при обучении нейросети, не ограничена обычной классическим МНК и может быть робастной к выбросам в данных [4], может включать в себя дополнительные слагаемые, например, регуляризующие решение.
- Построение нелинейных главных компонент нейросетью с "узким горлом" [2,3].
- При недостаточности линейных главных компонент для описания данных с нужной точностью с целью их дальнейшей визуализации в пространстве двух или трех первых ГК, возможна визуализация в пространстве нейросетевых нелинейных главных компонент или путем проекции на двумерное нелинейное многообразие, порождаемое картой Кохонена (набором квантующих данные ядер, между которыми задано топологическое соседство и на которые затем натягивается кусочно-линейная или гладкая интерполирующая поверхность).

Рост объемов баз данных в технике, бизнесе, медицине, экологии и растущие требования к точности решения ставят новые требования перед нейроимитаторами. Современные нейропакеты (NeuroSolutions, Statistica Neural Networks, Trajan и подобные) по прежнему ориентированы в основном на жесткую схему "создание нейросети → обучение → выдача прогноза-решения", что уже неадекватно. К тому же, наблюдается значительное отставание между давно ставшими стандартными в литературе передовыми методами и их реализацией в распространенных нейропакетах. В последних до сих пор реализованы в основном базовые методы, ныне трактуемые только как учебные, и не позволяющие эффективно решать современные задачи.

Автор разрабатывает собственное нейросетевое программное обеспечение с 1997г. В 1999-2000гг. начальная версия авторской программы NeuroPro была одной из наиболее широко используемых в России нейропрограмм. С тех пор опыт автора в решении индустриальных задач экологического и биомедицинского прогнозирования, обработки заводской технологической информации, обработки приходящей информации с многоканальных датчиков и принятия решения в реальном времени и

других задач потребовал значительного расширения программы как в плане реализации нейросетевых и иных методов обработки данных, так и в плане сервисных возможностей, автоматизирующих основные ритуалы действий.

Настоящая работа представляет собой нечто вроде эссе без стремления дать детальный список литературы как по нейромоделированию и нейроинформатике, так и для обоснования выдвигаемых авторских утверждений.

Методологические вопросы нейромоделирования

В разделе рассмотрение будет касаться только многослойных персептронов, решающих задачи прогнозирования (регрессии, авторегрессии, автоассоциации) и классификации с учителем. Несмотря на то, что оптимальная структура нейросети и настройки алгоритма обучения подбираются для достижения минимума ошибки обобщения – ошибки решения на независимой тестовой выборке, задача доказательства адекватности построенной нейромодели остается необходимой в любом случае. Регуляризующие критерии, при обучении штрафующие за излишнюю сложность модели, или критерии типа "бритвы Оккама", выбирающие баланс между сложностью и ошибкой модели, задачу обеспечения адекватности решают не полностью.

Современные нейропакеты не имеют средств для решения этой задачи. Часто пользователи даже не оценивают работу модели на независимой тестовой выборке.

Автор использует классический анализ коэффициентов корреляции между входными и внутренними сигналами сети и ошибками нейросети на обучающей и тестовой выборках. Для доказательства полезности подхода рассмотрим возможные причины, приводящие к возникновению индикаторов – высоких корреляций.

- Малые корреляции на обучающей и большие на тестовой выборках: причиной может быть недостаточность размера обучающей выборки и произошедшая во время обучения настройка на выборку.
- Высокие корреляции для обоих выборок: возможно, нейросеть недостаточна по объему (недостаточно гибка) или неоптимальны настройки алгоритма обучения. Возможно, потенциально достижимы меньшие ошибки обучения и обобщения.
- Наличие корреляций между сигналами внутри слоя сети означает либо избыточность архитектуры сети, либо, наоборот, какие-то внутренние ограничения (вынуждающие дублировать цепочки вычислений внутри сети).

Также возможна и двумерная визуализация точек выборки в координатах "входной сигнал – ошибка сети", позволяющая для задач прогнозирования обнаружить нелинейные зависимости между этими переменными. Количественным выражением силы нелинейной связи между переменными может быть значение кросс-энтропии. Для задачи классификации возможна двумерная визуализация в пространстве пар независимых признаков с раскраской точек в зависимости от правильности решения соответствующего примера выборки.

Таким образом, разработанные в рамках статистического и эмпирического подходов к анализу данных методы могут и должны применяться при нейромоделировании. Наиболее интересным и важным фактором здесь выступает то, что в роли переменных (дополнительных к наборам независимых и зависимых переменных самой задачи) можно использовать внутренние сигналы нейросети, и с их использованием детально анализировать выстроенный нейросетью при обучении алгоритм решения неформализованной задачи и целенаправленно ликвидировать узкие места и получать нужные свойства решения.

Наличие средств, автоматизирующих процесс выбора оптимальной структуры нейросети и настроек алгоритма обучения (наподобие реализованных в Statistica Neural Networks), часто бывает недостаточно, поскольку оптимизируется при этом величина

ошибки обобщения, но не свойства распределения ошибок модели. Поэтому в случае выбросов в данных не будет ни использовано робастных методов, ни сделано попытки откорректировать выбросы (хотя-бы путем исключения таких примеров из обучающей выборки). Все это может привести к переобучению – запоминанию отдельных свойств выборки или нахождению существующих только в пределах обучающей выборки корреляций между реализациями "сигнал+шум" для независимых и зависимых переменных. Очевидно, что качество обобщения это не повысит.

Ритуалы нейросетевой обработки данных

Большое число содержательно интересных задач обработки данных требует выполнения не одного, а нескольких, зачастую различающихся методами и целями шагов нейромоделирования. Это приводит к необходимости построения "ритуалов" решения таких задач. Так, для задач прогнозирования (регрессии) или классификации, решаемых на основе таблиц данных, схематичный ритуал был дан, например, в [5]. Для задач прогнозирования временных рядов нужны дополнительные или замещающие шаги ритуала – например, для определения глубины погружения ряда.

Ритуалы в нейроинформатике обычно конструируются так, чтобы обеспечивать решение задачи нейросетевыми методами. Однако, зачастую с точки зрения полезности, скорости обработки данных, гибкости или потенциальной расширяемости оптимальнее применение иных методов, классических или неклассических.

Так, в [5] для определения потенциальной разрешимости задачи предлагается определять внутреннюю размерность набора независимых переменных и внутреннюю размерность набора "независимые плюс зависимые переменные" и далее сравнивать эти две цифры. Равенство показателей указывает на возможную удачу при решении задачи (с предположением, что шум в данных позволит обеспечить нужную точность решения), превышение второго показателя над первым зачастую говорит о том, что для описания точек в более высокоразмерном пространстве недостаточно глобальной параметризации низкомерного пространства. Для надежности эмпирики внутреннюю размерность [5] предлагается определять по числу нейронов в "узком слое" нейросети-автоассоциатора, т.е. по числу нелинейных главных компонент.

Обучение нейросети-автоассоциатора требует порядка $O(N^3)$ операций при N примерах обучающей выборки и может быть неуспешным при наличии шумовых или малоинформативных признаков. Однако, можно использовать для оценки внутренней размерности данных фрактальную размерность, для рассчета которой к настоящему времени предложены алгоритмы со сложностью вплоть до $O(N)$ [6].

Вообще, стоит обратить внимание на современные передовые алгоритмы для задач data mining, разрабатываемые для обработки очень больших массивов данных. По многим направлениям удалось добиться линейных зависимостей вычислительных затрат от числа примеров и переменных, как, например, в [7] для обнаружения нетипичных паттернов во временных рядах. Использование таких быстрых алгоритмов вместо более медленных на персональных компьютерах (в силу невозможности распараллеливания) нейросетевых алгоритмов может ускорить этап разведочного анализа данных и этап анализа реализуемости (решаемости задачи). Одновременное же использование нейросетевых и иных алгоритмов позволяет с разных сторон взглянуть на задачу и ее свойства, подтвердить полученные результаты результатами других методов. Автор всегда начинает моделирование с построения классических линейных (линейная регрессия, ARIMA, линейная разделяющая поверхность), квадратичных (байесов классификатор), локальных моделей (классификатор по K ближайшим соседям) и только после этого переходит к обучению нейросетей – классические модели задают уровень точности, ниже которого не должна опускаться более гибкая

нелинейная нейромодель, и покажут реальную пользу от использования более сложных методов по сравнению с более простыми при решении конкретной задачи.

Автор развивает собственные ритуалы моделирования для итерационной схемы над этапами "анализ данных – оптимизация способа предобработки данных – исследование динамики показателей в ходе обучения нейросети – исследование статических свойств обученной сети – исследование свойств выдаваемого сетью решения". На каждом этапе (как анализа данных, так и моделирования) возникающие переменные и показатели анализируются с привлечением базовых стандартных методов – визуализация, статистика и т.д.

Гибридное программное обеспечение, реализующее нейросетевые и иные методы обработки и анализа данных

Два предыдущих раздела автор старался показать пользу от совместного и дополняющего друг друга использования классических и неклассических методов анализа данных и моделирования наряду с нейронными сетями. Укажем еще на пару работ, использующих гибридизацию для повышения точности решения задач.

В [8] для задачи аппроксимации многомерных данных произведен синтез метода динамических ядер и метода главных компонент. С каждым ядром связывается свое пространство линейных главных компонент над относящимися к ядру-кластеризатору точками данных. Оптимизация положений ядер и перерасчет главных компонент идет так, чтобы минимизировать в итоге дисперсию длин проекций точек на ГК. Получаем возможность визуализации данных в пределах каждого кластера (при ограничении двумя линейными ГК), снижаем ошибку описания данных по сравнению с тем же числом глобальных главных компонент. Сходный по идее метод "мозаичной регрессии" [1] способен работать и при многозначных или разрывных функциях.

В [9] для увеличения горизонта прогнозирования временных рядов используется построение иерархии предикторов. Сначала строится хороший одношаговый предиктор и для каждой точки ряда делается прогноз на p шагов вперед итерированием этого предиктора: получаем p переменных – новых колонок в таблице данных. Для коррекции возможной систематической ошибки итерированного предиктора эти p переменных выступают в качестве независимых признаков для нового, векторного предиктора-корректора на q шагов, причем q может быть больше p.

Т.о., при решении задач кроме переменных из обрабатываемой таблицы данных возникают переменные, связанные с методами моделирования (внутренние и выходные сигналы модели, ошибки модели), возникают "разметки" таблицы данных метками классов-кластеров и т.д. Т.е. построение каждой новой модели расширяет пространство переменных новыми переменными, которые потенциально можно использовать наряду с исходными. Главное – предоставить такую возможность в рамках моделирующей программы. Примером подобной реализации может служить модуль работы с временными рядами пакета Statistica, где каждое преобразование переменной (вычитание тренда, взятие первых разностей, сглаживание и т.д.) порождает новый ряд данных, а каждая модель – ряд прогноза модели и ряд невязок.

В авторской программе NeuroPro принят и реализуется подобный подход. Надо отметить, что лобовое решение об одновременной доступности всех порожденных моделями "промежуточных" переменных в случае реализации будет только усложнять работу пользователя. Решение о предоставлении пользователю того или иного набора переменных для дальнейшей работы или, по необходимости, для явного промежуточного выбора, делается на основании анализа выбранного пользователем действия (команды меню) и/или свойств текущих активных моделей. Так, при построении схемы голосования или усреднения прогноза нескольких моделей

предварительный выбор каких-либо переменных не нужен, а для оценки качества работы полученного коллективного прогноза достаточно анализа связей между ошибкой прогноза и входными и внутренними переменными моделей коллектива.

Такое явное ограничение набора переменных происходит и при любом процессе построения иерархических моделей, например, при построении нелинейных главных компонент для независимых переменных задачи нейросетью-автоассоциатором и решении затем задачи распознавания в пространстве этих нелинейных главных компонент новой нейросетью-классификатором, в схемах типа boosting и т.д.

Вдобавок, сейчас наблюдается некоторый нейроренессанс – возвращение к использованию простых базовых нейроалгоритмов вместо сложных, требующих тонкой настройки параметров. Так, градиентное обучение с постоянным шагом вместо алгоритмов типа сопряженных градиентов (требующих оптимизации величины шага на каждой итерации обучения для получения более быстрой скорости сходимости) позволяет избавляться от проблемы переобучения (overfitting) и необходимости регуляризации в нейросетях с избыточным размером [10], причем оптимальная длина шага, обеспечивающая как быстроту обучения, так и отсутствие переобучения, может быть найдена быстро на основе допускающей автоматизацию эмпирики [11]. Т.о, во многом снижается необходимость в сопутствующих процедурах (таких, как ранняя остановка обучения, введение регуляризующего слагаемого в целевую функцию для предотвращения переобучения) и их настройках, и гибридные алгоритмы на основе нейросетей вполне могут быть реализованы как черный ящик.

Заключение

Представлен взгляд на необходимость тесной интеграции нейросетевых и иных методов обработки и анализа данных. Такая интеграция накладывает определенные требования к программной реализации нейроимитаторов или моделирующих программ, включающих и нейроалгоритмы. Причем необходимы как специальная внутренняя организация программы для возможности гибкой и гладкой "стыковки" различных моделей и методов друг с другом, так и специальная организация интерфейса для простоты работы пользователя.

Литература

1. *Горбань А.Н., Россиев Д.А.* Нейронные сети на персональном компьютере. Новосибирск: Наука, 1996. - 276с.
2. *Ежов А.А., Шумский С.А.* Нейрокомпьютинг и его приложения в экономике и бизнесе. М.: МИФИ, 1998. - 224с.
3. *DeMers D., Cottrell G.* Non-linear dimensionality reduction / Advances in Neural Information Processing Systems 5 (1992). Morgan Kaufmann, 1993. – pp.580-587.
4. *Царегородцев В.Г.* Робастная целевая функция для задач нейроаппроксимации / Материалы настоящего семинара.
5. *Миркес Е.М.* Нейросетевые ритуалы анализа таблиц данных / VIII Всеросс. семинар "Нейроинформатика и ее приложения", 2000. Красноярск: КГТУ, 2000. - 204с. – С.119-120.
6. *Traina C. Jr., Traina A., Wu L., Faloutsos C.* Fast feature selection using fractal dimension / Proc. XV Brazilian Database Symposium, 2000. – 15p.
7. *Keogh E., Lonardi S., Chiu B.* Finding surprising patterns in a time series database in linear time and space / Proc. SIGKDD'02, Edmonton, Alberta, Canada, 2002. – 11p.
8. *Kambhatla N., Leen T.* Fast nonlinear dimension reduction / Advances in Neural Information Processing Systems 6 (1993). Morgan Kaufmann, 1994. – pp.152-157.
9. *Judd K., Small M.* Towards long-term prediction / Physica D, 2000, №136. – pp.31-44.
10. *Lawrence S., Giles C.L.* Overfitting and neural networks: Conjugate gradient and backpropagation / Proc. IHCNN'2000, Como, Italy, 2000. – pp.114-119.
11. *Wilson D.R., Martinez T.R.* The need for small learning rates on large problems / Proc. IJCNN'2001, Washington, DC, USA, 2001. – pp.115-119.